Sequence Compaction for Probabilistic Analysis of Finite-State Machines*

Diana Marculescu, Radu Marculescu, Massoud Pedram

Department of Electrical Engineering - Systems
University of Southern California, Los Angeles, CA 90089

Abstract - The objective of this paper is to provide an effective technique for accurate modeling of the external input sequences that affect the behavior of Finite State Machines (FSMs). The proposed approach relies on adaptive modeling of binary input streams as Markov sources of fixed-order. The input model itself is derived through a one-pass traversal of the input sequence and can be used to generate an equivalent sequence, much shorter in length compared to the original sequence. The compacted sequence can be subsequently used with any available simulator to derive the steady-state and transition probabilities, and the total power consumption in the target circuit. As the results demonstrate, large compaction ratios of orders of magnitude can be obtained without a significant loss (less than 3% on average) in the accuracy of estimated values.

I. Introduction

In the last decade, probabilistic approaches have received a lot of attention as a viable alternative to deterministic techniques for analyzing complex digital systems. In particular, the behavior of FSMs has been investigated using concepts from the Markov chain theory. Studying the behavior of the Markov chain provides us with different variables of interest of the original FSM. In this direction, [1][2] are excellent references where steady-state and transition probabilities (as variables of interest) can be successfully estimated in large FSMs. Both techniques are analytical in nature and resort to some simplifying assumptions, temporal independence on the primary inputs being the most notable one. These assumptions, however, limit the applicability and usefulness of the results. As a consequence, only logic simulation of the actual set of inputs can finally assert the accuracy of results.

It is, however, impractical to simulate long sequences of vectors, mostly when the target circuit is large or when many runs are needed to evaluate a number of alternative designs. From this perspective, a short/compact sequence of stimuli - which is representative of the typical application data - would be desirable to speed-up the simulation. Differently stated, the question to be answered is: having a sequence S_1, assumed representative of the data applied to a target sequential circuit, can we produce a shorter sequence S_2 such that the steady-state and transition probabilities of the signal lines are nearly preserved?

The aim of this paper is to address this issue and, based on a new Markov model, to propose an effective way to solve it not only for standard FSMs, but also for interacting FSMs. The knowledge of steady-state and transition probabilities is a very important topic by itself because both of them completely characterize the FSM behavior. However, as a particular domain where they have an immediate application, we chose the power estimation area. Without loss of generality, we will consequently emphasize the applicability of the new results on sequence compaction for power estimation.

Generating a minimal-length sequence of input vectors that satisfies a prescribed set of statistics in not a trivial task. Two effective techniques were recently presented in [3] [4] where the authors succeed in compacting large sequences with very small loss in accuracy. However, these approaches are suited only for combinational circuits and consider only first-order temporal effects (i.e. pairs of consecutive vectors) to perform sequence compaction. As we will prove in this paper, in the case of FSMs, this is insufficient for accurate estimation of transition probabilities. Temporal correlations longer than one time step can affect the overall behavior of the FSM and therefore, result in very different power consumptions. Let us illustrate this point using a simple example.

Example 1: Let S_1 and S_2 be two 4-bit sequences of length 26, as shown in Fig.1a. These two sequences have exactly the same set of first-order temporal statistics as shown in Fig. 1b. In this figure, we provide the wordwise transition graph for these two sequences. Each node in the graph represents to a distinct pattern which occurs in S_1 and S_2 (the topmost bit is the most significant one, e.g. in S_1, $v_1 = v_2 =$ '1', $v_3 =$ '2',..., $v_{26} =$ '9'). Each edge represents a valid transition between two patterns and has a nonzero probability associated with it. For instance, the pattern '13' in S_1 and S_2 is always followed by '5' (thus the edge between nodes '13' and '5' has the probability 1) whereas it is equally likely to have either '3' or '7' after '2' (thus the outgoing edges from node '3' have probability 0.5). We consider the graph in Fig.1b as a compact, canonical, characterization of sequences S_1 and S_2. Suppose now that S_1 and S_2 are input to the benchmark *s8* taken from the *mcnc'91* sequential suite. Looking at different internal nodes of the circuit, we see that the total number of transitions made by each node is very different when the circuit is simulated with S_1 or S_2. Moreover, the total power consumption at 20 MHz is 384µW and 476µW, respectively, showing a difference of more than 24% even for this small circuit. A natural question is then, why does this difference appear, in spite of the fact that S_1 and S_2 have the same characteristic graph plotted in Fig.1b.

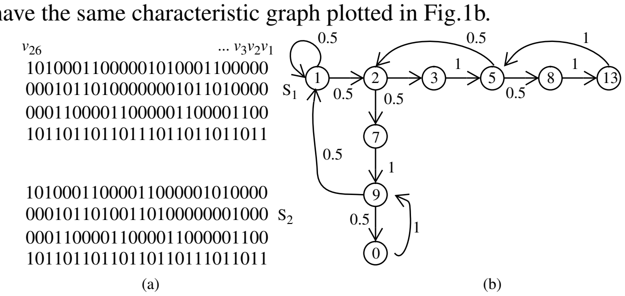

Fig.1: Two sequences with the same first-order statistics

The reason is that S_1 and S_2 have a different set of second-order statistics that is, the sets of triplets (three consecutive patterns) are different. For instance, the triplet (1,2,7) in S_2 does not occur in S_1; the same observation applies to the triplet (5,2,3) in S_2. The conclusion to note is that having the same set of one-step transition probabilities *does not* imply that the set of second-order or higher-order statistics are identical and, as it was just illustrated in this

*This research was supported by DARPA under contract F33615-95-C1627, SRC under contract 97-DJ-559, and NSF under contract MIP-9457392.

DAC 97, Anaheim, California
(c) 1997 ACM 0-89791-920-3/97/06 ..$3.50

small example, for FSMs higher order statistics can make a significant difference in total power consumption. The initial problem of compacting an initial input sequence so as to preserve the set of steady-state and transition probabilities of the FSM can be now cast in terms of power as follows: can we transform a given input sequence into a shorter one, such that the new body of data is a good approximation of the initial sequence as far as total power consumption is concerned?

Addressing these issues, the present paper improves the state-of-the-art in two ways. First, it shows the effect of finite-order statistics of the input sequence on FSMs behavior. Second, based on the vector compaction paradigm, it provides an original solution for power estimation problem in FSMs and interacting FSMs. Among the theoretical results provided here, three are noteworthy for probabilistic FSM analysis. First, under the stationarity and ergodicity assumptions, completely capturing the characteristics of the external inputs of the FSM is sufficient to jointly characterize the input and state lines. Second, if the sequence feeding the target circuit has order k, then a lag-k Markov chain model of the sequence will suffice to model correctly the joint transition probabilities of the primary inputs and internal states in the target circuit. Lastly, if the input sequence has order two or higher, then modeling it as a lag-one Markov Chain cannot exactly preserve the first-order joint transition probabilities (primary inputs and internal states) in the target circuit.

The foundation of our approach is probabilistic in nature; it relies on *adaptive* (*dynamic*) *modeling* of binary input streams as first-order and higher-order Markov sources of information. The adaptive modeling technique itself (best known as Dynamic Markov Chain or DMC modeling) was used very recently for power estimation [4]. However, this formulation is not completely satisfactory for our purpose; in order to capture high-order temporal effects, we thus extend the initial formulation to handle groups of more than two consecutive input vectors.

The paper is organized as follows: section II formalizes the power-oriented vector compaction problem. Next, based on Markovian information sources, we present in section III the main results about the effect of finite-order statistics on FSM and interacting FSM behavior. Section IV introduces a DMC-based procedure for vector compaction. In section V we present some experimental results and finally, we conclude by summarizing our main contribution.

II. Data Compaction For Power Estimation

Assuming that a gate level implementation is available, one can estimate the total power dissipation by summing over all the gates in the circuit the average power dissipation due to the capacitive switching currents, that is: $P_{avg} = \frac{f_{clk}}{2} \cdot V_{DD}^2 \cdot \sum_n (C_n \cdot sw_n)$

where f_{clk} is the clock frequency, V_{DD} is the supply voltage, C_n and sw_n are the capacitance and the average switching activity of gate n, respectively. From here, the average switching activity per node is the key parameter that needs to be correctly determined. However, this parameter is highly sensitive to the input statistics, namely it depends significantly on transition probabilities among different signal lines. As shown in the previous section, high-order information sources make a significant difference in power consumption for sequential machines.

The *vector compaction problem* for FSMs is formulated as follows: for any sequence of length L_0, find another sequence of length $L \ll L_0$ (consisting of a subset of vectors from the original sequence), such that the average joint transition probability on the primary inputs and present state lines is preserved *wordwise,* for *k+1* consecutive time steps. More formally, the following holds:

$$\left| \begin{array}{l} p(x_n s_n x_{n-1} s_{n-1} \ldots x_{n-k} s_{n-k}) - \\ - p'(x_n s_n x_{n-1} s_{n-1} \ldots x_{n-k} s_{n-k}) \end{array} \right| < \varepsilon \qquad (1)$$

where p and p' are the probabilities in the original and compacted sequences, respectively. This condition simply requires that the joint transition probability for inputs and states ($x_i s_i$) is preserved within a given level of error for $k + 1$ consecutive time steps. Before going further, we note the particular case when $k = 1$, which is the theoretical basis of vector compaction techniques recently published in [3][4].

III. Markovian Sources Of Information

A. Finite-order memory models

Without loss of generality, we restrict ourselves to finite binary strings, that is, finite sequences consisting only of 0's and 1's. The set of events of interest is the set S of all finite binary sequences on b bits. A particular sequence S_1 in S consists of vectors $v_1, v_2, \ldots, v_n$ (which may be distinct or not), each having a positive occurrence probability. An attractive subclass of information sources is the class of Markov sources which can be conveniently modeled as Markov chains of finite-order.

Definition 1. (lag-k Markov chain) A discrete stochastic process $\{v_n\}_{n \geq 1}$ is said to be a lag-k Markov chain if at any time step $n \geq k+1$: $p(v_n | v_{n-1} v_{n-2} \ldots v_1) = p(v_n | v_{n-1} v_{n-2} \ldots v_{n-k})$ (1)

In particular, any lag-one Markov source, is characterized by the set of states (nodes in the corresponding graph representation) and the set of transition probabilities p_{ij} from state v_i to the next state v_j. We note that any lag-k Markov chain can be reduced to a lag-one Markov chain using the following (all proofs are in [8]):

Proposition 1. If $\{u_n\}_{n \geq 1}$ is a lag-k Markov chain then $\{v_n\}_{n \geq 1}$, where $v_n = (u_n, u_{n+1}, \ldots, u_{n+k-1})$, is a multivariate first-order Markov chain.

B. The effect of finite-order statistics on FSM behavior

Now we turn our attention from the input sequence to the circuit and investigate the effect of input statistics on the transition probabilities (primary inputs and present state lines) in the target circuit. As shown in Fig.2, we model the 'tuple' (*input_sequence, FSM*) by the 'tuple' (*Markov_chain, FSM*), where *Markov_chain* models the *input_sequence* and *FSM* is the sequential machine where the transition probabilities have to be determined. In what follows, x_n, s_n will denote the inputs and states of the target sequential machine; $p(x_n s_n)$ is the probability that the input is x_n and the state is s_n at time step n.

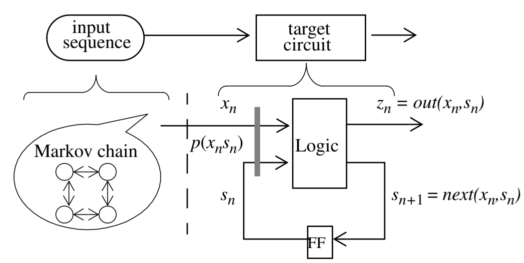

Fig.2: The tuple (*Markov-Chain, FSM*)

We are interested in defining the joint probabilities $p(x_n s_n)$ and $p(x_n s_n x_{n-1} s_{n-1})$ because, as we can see in Fig.2, they capture the characteristics of the input (primary inputs and present state lines) that feeds the next state and the output logic of the target circuit. Under the general assumptions of *stationarity* and *ergodicity*, we can prove the following result:

Theorem 1. If the input x_n applied to a target sequential circuit can be modeled by a lag-k Markov chain then, for any $n \geq k+1$ the following holds:

$$p(x_n s_{n-k} | x_{n-1} x_{n-2} \dots x_{n-k}) = \\ = p(x_n | x_{n-1} x_{n-2} \dots x_{n-k}) \cdot p(s_{n-k} | x_{n-1} x_{n-2} \dots x_{n-k}) \quad (2)$$

Theorem 2. If the sequence feeding a target sequential circuit has order k, then a lag-k Markov chain which correctly models the input sequence, also correctly models the k-step conditional probabilities of the primary inputs and internal states, that is $p(x_n s_n | x_{n-1} s_{n-1} x_{n-2} s_{n-2} \dots x_{n-k} s_{n-k}) = p(x_n | x_{n-1} x_{n-2} \dots x_{n-k})$.

We note therefore that preserving order-k statistics implies also that order-k statistics will be captured for inputs and states. In general, modeling a k-order source with a lower order model may introduce accumulative inaccuracies. From a practical point of view, this means that underestimating a high-order source, one may end up not preserving correctly even the first-order transition probabilities. In terms of power consumption, this will adversely affect the quality of the results. However, we will show later that increasing the order of the input model will decrease the error in correctly capturing the joint transition probabilities for inputs and states.

C.Interacting FSMs and high-order information sources

Modern designs where interacting finite state machines are present offer a good example where high-order information sources have found applicability. As presented in [5], the decomposition of large FSMs into smaller, interacting FSMs may be useful for both area and performance reasons. In practice, three options are available: *parallel decomposition* (both submachines are supplied with the same input sequence, but operate independently), *cascade decomposition* (one submachine has information about the internal state of the another one) and finally, a type of *complex decomposition* where each submachine is provided with information about the current state of the other submachine. Having on inputs a Markov source of order k, any of the aforementioned topologies may increase the order of the source at the output. However, we may assume a finite-order Markov source for the output, since for a given level of accuracy, there exists a general result that *guarantees* the existence of a *finite limit* in the resulting order:

Theorem 3. [6] Let $P = (p_{ij})_{1 \leq i, j \leq N}$ be the transition probability matrix of a lag-one Markov chain $\{x_n\}_{n \geq 1}$ with N states. If $p_{ij} > 0$ for any i, j and $\lambda = min_{i,j,k,l} \dfrac{p_{kj} \cdot p_{il}}{N^2 \cdot p_{ij} \cdot p_{kl}}$, then for any arbitrary function $z_n = f(x_n)$ the following holds $\forall\ k$ and $x_{n-k-1} \neq x'_{n-k-1}$[1]:

$$\left| \begin{matrix} p(z_n | z_{n-1} \dots z_{n-k} x_{n-k-1}) - \\ -p(z_n | z_{n-1} \dots z_{n-k} x'_{n-k-1}) \end{matrix} \right| \leq (1-\lambda)^k \quad (3)$$

In other words, this theorem states that even if the output is not of finite order, it can be approximated as such up to a bounded error. Based on this result, we can prove the following:

Corollary 2. Assume that the input of the FSM can be written as $x_n = f(w_n)$ where f is an arbitrary function and $\{w_n\}_{n \geq 1}$ is a lag-one Markov chain. If the order of the Markov model used to represent the input is increased, then the error for estimating the joint transition probabilities for inputs and states decreases.

Thus, the error of using a finite-order model for a non-finite order discrete process decreases exponentially with the order used. Hence, the larger the order, the better we approximate the model on the input and also the joint transition probabilities for inputs and states.

[1]It can also be shown that λ is less than one. The result may be extended to Markov chains of order greater than one.

IV. High-Order Dynamic Markov Models

Dynamic Markov Chain (DMC) technique was introduced in the literature of data compression few years ago and used recently to adaptively compact data for power simulators [4]. The structure DMT_1 used by authors in [4] is general enough to capture completely the correlations among all bits of the same input vector and also between two successive input patterns. However, it has conceptually no inherent limitation to be further extended to capture temporal dependencies of higher orders.

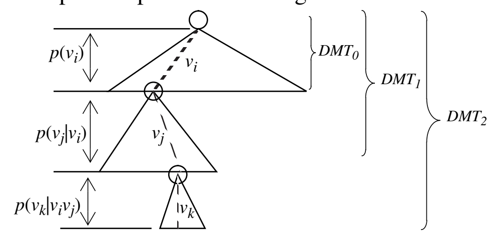

Fig.3: A high-order Markov tree

For instance, if we continue to define recursively DMT_2 (as a function of DMT_1), we can basically capture second-order temporal correlations. For any sequence where v_i, v_j, v_k are three consecutive vectors (that is, $v_i \rightarrow v_j \rightarrow v_k$), the tree DMT_2 looks like in Fig.3.

The following result, gives the theoretical basis for using the DMC technique to capture high-order temporal correlations.

Theorem 3. The general structure DMT_k and its parameters completely capture spatial and temporal correlations of order k.

In practice, we can imagine the following simple procedure for vector compaction: during a one-pass traversal of the original sequence (when we extract the bit-level statistics of each individual vector $v_1, v_2 \dots, v_n$ and those corresponding to $p \leq k+1$ consecutive vectors $(v_1 v_2 \dots v_p)$, $(v_2 v_3 \dots v_{p+1}), \dots)$ we grow simultaneously the tree DMT_k up to the end of the original sequence. This is followed by a generation phase driven by the user-specified compaction parameter *ratio* that is, a total of $m = n/ratio$ vectors are generated. The generation procedure uses a modified version of the *dynamic weighted selection algorithm* [7]. The pseudocode for the generation procedure and a detailed example can be found in [8]. We note that this strategy does note allow 'forbidden' vectors that is, those combinations that did not occur in the original sequence, will not appear in the final compacted sequence either. This is an essential capability needed to avoid 'hang-up' ('forbidden') states of the sequential circuit during simulation process for power estimation.

V. Experimental Results

The overall strategy is depicted in Fig.4.

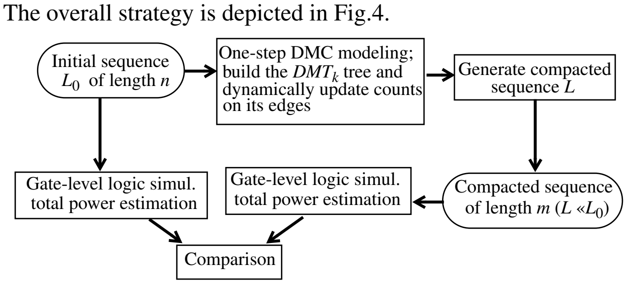

Fig.4: Experimental setup

We assume that the input data is given in the form of a sequence of binary vectors. Starting with an k-bit input sequence of length n, we perform a one-pass traversal of the original sequence and simultaneously build the basic tree DMT_k; during this process, the frequency counts on DMT_k's edges are dynamically updated. The next step in Fig.4 does the actual generation of the output sequence (of length m). If the initial sequence has the length n and the new generated sequence has the length $m < n$ then the outcome of this process is a compacted sequence, equivalent to the initial one as far as total power consumption is concerned; we say that a *compaction ratio* of $r = n/m$ was achieved. Finally, a validation step is included in the strategy; we have used an in-house gate-level logic simulator developed under SIS. The total power consumption of some *mcnc'91* and *ISCAS'89* benchmarks has been measured for the initial and the compacted sequences, making it possible to assess the effectiveness of the compaction procedure (under both zero- and real-delay models).

In Table 1, we provide only the real-delay power dissipation results for different initial sequences of 4,000 vectors for *mcnc'91* circuits and 10,000 vectors for *ISCAS'89* circuits. These sequences were produced using a second order information source based on the Fibonacci series. As shown in Table 1, the sequences were compacted with two different compaction ratios (namely $r = 5$ and 10) using two Markov models: one of order one and another one having order two. We give in this table the total power dissipation measured for the initial sequence (column 3) and for the compacted sequence using both models (columns 4-7). On a Sparc 20 workstation with 64 Mbytes of memory, the time necessary to read and compress data was less than 5 sec. for both models. Since the compaction with DMC modeling is linear in the number of nodes in the structure DMT_k, these time values are far less than the actual time needed to simulate the whole sequence. During these experiments, the number of nodes allowed in the Markov model was on average 10,000 for *mcnc'91* circuits and 200,000 for *ISCAS'89* circuits.

Table 1: Total Power (uW@20MHz) for sequences of order 2

Circuit	Inputs/ FFs	Power for initial seq.	Power for $r = 5$ Order 1	Power for $r = 5$ Order 2	Power for $r = 10$ Order 1	Power for $r = 10$ Order 2
bbara	4/4	747.10	838.12	748.22	866.76	744.99
dk17	2/3	1439.43	1281.30	1438.10	1250.20	1438.00
mc	3/2	295.84	212.39	291.11	196.76	287.85
planet	7/6	8517.14	4649.90	8046.20	3596.80	7565.50
shiftreg	1/3	144.60	115.26	144.22	109.73	143.84
s1196	14/18	7025.31	6842.34	7023.21	6668.36	6995.10
s1423	17/74	5624.64	5335.58	5557.52	5203.98	5489.51
s5378	35/164	13826.55	13576.21	13812.21	13304.25	13762.15
s820	18/5	4120.95	3839.72	4026.35	3668.74	4301.42
s9234	36/211	12531.45	12796.41	12334.32	13037.15	12271.23
		Avg.% err.	14.55	1.29	17.47	2.44

As we can see, for the model of order 2, the quality of results is very good even when the length of the initial sequence is reduced by one order of magnitude. Thus, for *bbara* in Table 1, instead of simulating 4,000 vectors with an exact power of 747.10µW, one can use only 800 vectors ($r = 5$) with an estimate of 748.22µW or just 400 vectors ($r = 10$) with power consumption estimated as 744.99µW. This reduction in the sequence length has a significant impact on speeding-up the simulative approaches where the running time is proportional to the length of the sequence which must be simulated. On the other side, using a first-order model, the quality of the results can be seriously impaired. For instance, in the case of benchmark *plane*t, we can erroneously predict a total power of 3596.80µW (57.78% error) if $r = 10$. This is because for a sequence generated with a second-order source, a model that considers only pairs of two consecutive vectors cannot preserve correctly even the first-order transition probabilities for the primary inputs and state lines.

We also studied the sensitivity of the proposed approach to the choice of initial seeds used for random excitation of the DMC model. Using different seeds for the random number generator, we run a set of 1,000 experiments for the DMC technique. In almost all cases, the second-order model yielded errors less than 5% compared to the exact simulation. On the other hand, using a first-order model significantly impaired the accuracy of the results: for some circuits, more than 80% of the runs produced results with more than 10% error compared to the original sequence.

To assess the importance of correctly modeling the input sequence, we give in Table 2 our results for cascade and complex configurations with a compaction ratio of 5. In the first case we cascaded benchmarks *ex4* (from *mcnc'91* suite) and *s1196* (from *ISCAS'89* suite) and we estimated the total power consumption for both of them. In the second case, we used a complex topology where benchmarks *ex3* and *planet* interact. Looking at the results in Table 2 we can conclude that only the second order model is appropriate for this type of analysis.

Table 2: Total Power (uW@20MHz) for sequences of order 2 for interacting FSMs

Configuration	Inputs/FFs	Power for initial seq.	Power for order 1	Power for order 2
cascade	6/22	5762.03	6158.38	5772.68
interacting	6/10	11278.65	10290.09	11188.82
		Avg.% err.	7.82	0.49

We note that using a lower order model than needed may also significantly impair our ability to correctly estimate the switching activity in a *node-by-node* analysis. Typical results are given in [8].

VI. Conclusion

In this paper we investigated from a probabilistic point of view the effect of finite-order statistics of the input sequence on FSM and interacting FSM behavior. Based on dynamic Markov modeling, we proposed an effective approach to compress an initial sequence into a much shorter one such that the steady state and transition probabilities (and therefore the total power consumption) in the target circuit are preserved.

The mathematical foundation of this approach relies on adaptive modeling of binary input streams as first- and higher-order Markov sources of information. For the first time to our knowledge, the effect of temporal correlations longer that one clock-cycle on the power dissipation in FSMs and networks of interacting FSMs was studied. As shown by the experimental results, large compaction ratios can be obtained with less than 3% loss in accuracy for total and node-by-node power consumption.

The results presented in this paper represent an important step towards understanding the FSM behavior from a probabilistic point of view.

References

[1] G. Hachtel, E. Macii, A. Pardo, and F. Somenzi, 'Probabilistic Analysis of Large Finite State Machines', in *Proc. ACM/IEEE Design Automation Conference*, pp. 270-275, June 1994.

[2] C.-Y. Tsui, J. Monteiro, M. Pedram, S. Devadas, A. M. Despain, and B. Lin, 'Power Estimation Methods for Sequential Logic Circuits', in *IEEE Trans. on VLSI Systems*, vol.3, no.3, Sept. 1995.

[3] D. Marculescu, R. Marculescu, and M. Pedram, 'Stochastic Sequential Machine Synthesis Targeting Constrained Sequence Generation', in *Proc. ACM/IEEE Design Automation Conference*, pp. 696-701, June 1996.

[4] R. Marculescu, D. Marculescu, and M. Pedram, 'Adaptive Models for Input Data Compaction for Power Simulators', in *Proc. Asia and South-Pacific Design Automation Conference*, pp. 391-396, Japan, Jan. 1997.

[5] S. Devadas and A.R. Newton, 'Decomposition and factorization of Sequential Finite State Machines', in *IEEE Trans. on Computer*-Aided Design of Integrated Circuits, vol.8, No.11, pp. 1206-1217, Nov. 1989.

[6] T.E. Harris, 'On Chains of Infinite Order', in *Pacific J. Math.*, vol. 5, pp. 707-724, 1955.

[7] J.W. Green and K.J. Supowit, 'Simulated Annealing without Rejected Moves', in *Digest. of Intl. Conference on Computer Design,* pp. 658-663, Oct. 1984.

[8] D. Marculescu, R. Marculescu, and M. Pedram, 'FSM Analysis Using High-Order Markov Models', Technical Report CENG 97-08, Univ. of Southern California, Oct. 1996.